Personalized information retrieval models integrating the user's profile

Chahrazed Bouhini, Mathias Géry, Christine Largeron

Personalized Information Retrieval Models Integrating the User's Profile

Chahrazed Bouhini*†, Mathias Géry*, Christine Largeron*
*Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d Optique Graduate School,
Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France
{ chahrazed.bouhini, mathias.gery, christine.largeron }@univ-st-etienne.fr
† Insight Centre for Data Analytics, University College Cork, Ireland
† chahrazed.bouhini@insight-centre.org

***Abstract*—Personalized Information Retrieval (PIR) exploits the user's data in order to refine the retrieval, like for instance when users with different backgrounds may express different information needs with the same query. However, this additional source of information is not supported by the classical Information Retrieval (IR) process. In order to overcome this limit, we propose to generate the user profile out from his profile and social data. Then, we introduce several Personalized Information Retrieval models which integrate this profile at the querying step, allowing to personalize the search results. We study several combinations of the initial user's query with his profile. Furthermore, we present a PIR test collection that we built from the social bookmarking network Delicious, in order to evaluate our PIR models. Our experiments showed that the PIR models improve the retrieval results.**

***Index Terms*—Personalized Information Retrieval, User's Profile, Social Data, Folksonomies**

I. Introduction

Classical information retrieval systems are based on the keyword search: given a collection of resources and a user information need, they aim to provide relevant resources to the user. Most of the time, this information need is expressed with a query composed of a few keywords. Usually, less than three words, because few users are able to formulate their needs with complex queries [1], [2], [3]. However, it has been shown that the queries can be ambiguous since users with different needs can use the same query even if they expect different relevant resources [4]. A solution to enhance the information retrieval process, without altering the way that the users specify their requests, is a personalized information retrieval which takes into account not only the query but also other information given, directly or not, by the user [5]. The advent of the Web 2.0, with social tagging systems which allow the users to annotate resources, has notably provided complementary information useful to disambiguate the query and personalize the search. This lead notably to Personalized Information Retrieval (PIR) which can be defined as the incorporation of information related to social networks into the information retrieval process [6]. To this end, the main task of Personalized Information Retrieval is to exploit the user's profile and integrate it during the IR process. This requires to adapt the usual IR models and systems in order to deal with this user's social data.

In this article, we propose to generate the user's profile from his social annotations. More specifically, once the user's profile is generated, we investigate the way to exploit the information provided by the term distribution within the user's profile. Then, we present social IR models called $BM25S_{profile}$, $BM25S_{ScoreComb}$ and $BM25S_{FreqComb}$, which integrate this user's profile during the querying step.

In order to evaluate our social IR models that take in consideration the users' social data, we built a social IR test collection providing a user-centered data (user-centered relevance judgments). This collection is based on the social annotations extracted from the social collaborative bookmarking system **Delicious**[1].

This article is organized as follows: we discuss some related works in Section II and we explain the motivations of our work in Section III. In Section IV, we present our methodological framework by describing the approach that we used to generate the user's profile and we detail the main contribution of our work which consists in integrating this user's profile in the querying step. Further, we present in Section V the results of the experiments conducted on the social test collection we built, described and we conclude with perspectives in Section VI.

II. Related Work

Personalized Information Retrieval (PIR) attempts to extend classical IR by taking into consideration the user's profile within his social network. This profile may be integrated in order to refine the query, like in query expansion, or during the indexation and the ranking of the documents. In this last case, considered in this article, the ranking of a document depends not only on the matching between the document and the query, like in classical IR, but also on the matching between the user's interest and the document.

A. User's profiles

The user's interests are generally summarized by his profile, which may be built from his social annotations. These annotations correspond to short descriptions (or tags) given by the

[1] https://delicious.com/

user to the resources. The user's profile can also integrate the tags used by the user's neighbors in his social network. In several models, notably inspired by the Vector Space Model, the document, the query and the profile are then described by vectors defined in the same space: the terms. The matching between a query and a document as well as the matching between the user's profile and the document are computed with the cosine similarity. This approach has been proposed in different works [7], [8], [9], [10].

B. Tags based IR

Diederich and Iofciu propose an approach where the search exploits the tags put on the document by the users [7], but the search is not really personalized because the same set of documents is returned to the users who gave the same query, whatever their profiles. Indeed, the ranking of the document depends on the similarity between the query and the tag-based description of the document and it is not based on the appearance of the query terms into the document like in a classical IR process.

C. Combining tags and profiles for Personalized IR

The vector space model has been adapted by Xu *et al.* for personalized IR [8]. As explained previously, the query, the document and the user's profile are represented by vectors in a space defined by the tags appearing in the annotations. The weight of a tag in the vectors associated to the user's profile and the resource are computed with the usual TF.IDF formula. Then, two scores are computed using the cosine, on the one hand the similarity between the document and the query and, on the other hand, the similarity between the document and the user's profile. Each score is used to generate a ranked document list, the first one in function of the query and the second one in function of the user's profile. The relevance of the document is finally estimated by a global score defined as a linear combination of the previous scores where the Weighted Borda Fuse algorithm (WBF) is used to aggregate the rankings.

The personalized model of Xu *et al.* [8] has been adapted by Vallet *et al.* [10] and by Cai and Li [9]. In the variant introduced by Vallet *et al.*, the similarity between a user and a document is computed with BM25 instead of TF.IDF. The difference between the proposition of Cai and Li and the model of Xu *et al.* lies in the combination of the query score and of the user score: the former work proposes to compute the final score as the product of these two scores, when the latter proposes to obtain it by an aggregation of the rankings. Moreover, in the approach of Cai and Li, the vectorial representation of the resource differs slightly from the one adopted by Xu *et al.*.

Another model has been proposed by Kashyap *et al.*, based on a linear combination of the ranking given by a classical search engine and the social ranking [11]. However, in this model, the social ranking is done using an adaptation of the PageRank on a graph representing the relationships between the documents, the queries and the users or groups of users. If the graph is a rich model of the interactions between the different components of a PIR, it is also more complex to handle.

TABLE I
EXAMPLE: QUERY, DOCUMENTS AND USER'S PROFILES.

	t_1 = smartphone	t_2 = android
Query q	1	1
Document d_1	1	0
Document d_2	0	1
Alice	2	1
Bob	1	2

D. Discussion

From these previous works, we retain the idea that it is important to exploit the social annotations of the user. So in our models, we propose also to combine the social data and the query data. However, we consider that the representation of the documents can not be based only on the tags like in the previously mentioned works. Moreover, we think that combining the document content with this user's profile requires IR techniques that are able to handle large textual documents. This led us to introduce an original approach presented in the following sections.

III. PERSONALIZED IR EXPLOITING FOLKSONOMIES

Our approach is based on several assumptions as detailed below.

A. Information need: query and personal interests

Almost all test collections, in IR research, assume that queries have a single interpretation representing the information need expressed by one user, which is implicitly defined in his relevance judgments [4]. However, in practice this is not necessary the case. For this reason, in this paper we propose a framework for Personalized Information Retrieval based on folksonomies. Such a system should be able to handle ambiguous queries, i.e. queries having potentially several interpretations representing different information needs.

For example, suppose that two users Alice and Bob ask the same query $q =$ "smartphone android" (cf. Table I).

We consider two documents d_1 and d_2; each document contains one query term, but *smartphone* is more important than *android* in the first document since d_1 contains only *smartphone*, and *android* is more important than *smartphone* in the second one since d_2 contains only *android*. Assuming that the two query terms have the same importance, a classical IR system should estimate that d_1 is equally relevant as d_2 for the query "*smartphone android*". However, depending on the user and his personal interests, the information need behind this query may focus either on the term *smartphone* or on the term *android*.

Now, if we suppose that Alice is mainly interested in smartphone devices, then her information need is probably centered around smartphones with an opening on Android, and thus the query term *smartphone* should be more important than the query term *android*. On the other hand, if Bob is mainly interested by the Android operating system, then his information need is probably centered around Android, and thus the query term *android* should be more important than the query term *smartphone*.

A personalized information retrieval system should be able to identify the user's personal interests, in order to better interpret the information need behind his queries, and returns lists of relevant documents to the users depending on their personal interests. In our example, a Personalized Social IR system should consider d_1 as more relevant than d_2 for Alice and the opposite for Bob.

B. Folksonomies and user's profile

We assume that folksonomies may be exploited in order to build the informational social profile of the user that could represent the user's interests and that could help the system to handle ambiguous queries and return personalized results to the user. As pointed out in related literature, the user's profile can be inferred from his social annotations [12], [13].

C. Integrating user's profile

We think that the integration of the user's profile within the IR model is an important part of the personalization approach.

Since we exploit the social information about the user to generate his profile, we assume that the important terms representing the user's interests should appear in this profile. Thus, reweighting such important terms when they are found within the document, should improve the document relevance score and allow to return the personalized relevant documents.

Then, the aim of our approach is to exploit together several sources of information:

- The textual content of the documents, containing thousand of terms;
- The explicit user's information need, expressed as a short query composed of a few terms;
- The implicit user's information need, which is somewhere expressed by his profile.

D. Weighting large textual documents

One aim of this work is to handle textual documents containing thousands of terms, unlike most related works which only handle small sets of tags describing the document. For this reason, a basic *tf.idf* weighting function is not suitable: we need to use term weighting function that is designed to process large textual documents and the well-known Okapi BM25 weighting function seems to be one of the best choice.

The behavior of such non-binary functions, on the document side as well as on the query side, aims at giving less importance to each new occurrence of a term in the same document: there is a significant difference if a term appears two times in the same document (and then the increase of its weight should be significant), but there is almost no difference between 1000 and 1001 occurrences of a term in the same document (and then the increase of its weight should be negligible). This behavior is called ”saturation” [14].

E. Weighting short queries and user's profile

On the query side, in classical information retrieval, the terms have usually a binary weight: the term appears in the query (term frequency = 1) or the term does not appear (term frequency = 0).

In the case of profiles, including several occurrences of the same term, it seems interesting to consider the term frequencies, like weighting functions do on the document side. Especially, it seems important to provide the saturation effect of a non-linear weighting function.

F. Combining binary queries with user's profile

Combining the user's profile (term frequencies) with a query (binary) and then computing the matching score with textual documents containing thousands of terms raises different issues than computing the matching between binary queries and textual documents (classical IR) or than only combining user's profile and the document's profile represented by a set of tags, like for instance in the work of Cai and Li [9].

Especially, combining binary query with user's profile raises an important issue: how to equilibrate the importance of these two different kind of information?

In this work, we propose to study four different strategies:

- the first one (called classical $BM25$) considers only the explicit user's information need, expressed as a short binary query composed of a few terms;
- the second one (called $BM25S_{profile}$) considers only the implicit user's information need, expressed by its non-binary profile composed of many terms;
- the third one (called $BM25S_{ScoreComb}$) combines the initial binary queries and the user's profile at a scoring level;
- the fourth one (called $BM25S_{FreqComb}$) combines the initial binary queries and the user's profile at a term frequencies level.

Our approach is inspired by the work of Robertson *et al.,* who have shown, while combining several textual fields that compose a structured document, that the ”term frequencies level” strategy is theoretically and experimentally better than the ”scoring level” strategy [15]. One aim of this work is to check this conclusion in the context of PIR.

IV. Personalized Information Retrieval models

We present the well known Information Retrieval model BM25 (cf. Section IV-B), on which are based our Personalized Information Retrieval models, as well as our three PIR models that exploit the user's profile in order to refine the user's query:

- $BM25S_{profile}(d, u)$ (PIR model): this model returns a ranked list of documents that are relevant for a given user u considering only his profile. Thus, the binary query q is simply replaced by u in equation (6).

- $BM25S_{ScoreComb}(d, q, u)$ (combined PIR model): this model returns a ranked list of documents that are relevant for a given user u considering his binary query q combined **at the scoring level** with his profile.
- $BM25S_{FreqComb}(d, q, u)$ (combined PIR model): this model returns a ranked list of documents that are relevant for a given user u considering his binary query q combined **at the term frequencies level** with his profile, as recommended by Robertson *et al.* [15].

A. Notations

We represent the "social tagging data", also known as *Folksonomies* [10], by a tuple $\mathcal{R}_S = < U,\ T,\ D,\ A,\ Rel >$, where:

- U is a set of social network users;
- T is a set of index terms;
- D is a set of documents on the Web (images, videos, Web pages, etc.);
- A is a set of social annotations;
- and $Rel \subseteq U \times U$ is a set of relationships between pairs of users.

Rel is such that $(u, u_y) \in Rel$ if there is a social relationship between the users u and u_y. The users related to u are typically those declared explicitly by u as his $neighbors$ where $neighborhood(u) = \{u_y\ /\ (u, u_y) \in Rel\}$.

We note also that:

- a document $d \in D$ is represented by a set of terms $t \in T$ and a term t may appear one or several times in d;
- $tf_{d,t}$ is the term frequency of t in d;
- $w_{d,t}$ is the weight of a term t for a document d, computed using its term frequency $tf_{d,t}$;
- $a_z \in A$ with $a_z = < d, u, T_z >$ is the annotation of the user u for the document d using a subset of terms $T_z \subset T$.

On the query side, we note that:

- Q is a set of users' queries;
- each query $q \in Q$ is represented by a set of terms $t \in T$;
- $Qrels$ is a set of global relevance judgments (i.e. independently of the user);
- $qrels_q \subset D$ denotes the set of relevant documents for q independently of the user;
- Q_{UC} is a set of couples (q, u) where the query q is issued by the user u to express his personal information needs;
- $Qrels_{UC}$ is a set of user-centered relevance judgments (i.e. dependently to a given user);
- $qrels_{q,u} \subset D$ is the set of relevant documents for the query q and the user u.

B. $BM25$: classical Information Retrieval weighting function

We choose to build our Personalized Information Retrieval models on the IR weighting function BM25 [16], which is one of the most used indexation models in the IR research benchmarks such as INEX[2], TREC[3], etc. This IR weighting function is composed by three parts:

- Aboutness: $TF_{d,t}$ estimates the aboutness of the document d considering the term t (cf. equation 1).
- Discrimination: DF_t estimates the discriminating power of the term t to the documents (cf. equation 2).
- Query terms: $QTF_{q,t}$ estimates the importance of the term t within the query q (cf. equation 3).

Aboutness ($TF_{d,t}$), term discriminating power (DF_t) and query terms importance ($QTF_{q,t}$) are defined as follows [16]:

$$TF_{d,t} = \frac{(k_1 + 1) \times tf_{d,t}}{k_1 \times \left((b - 1) + b \times \left(\frac{dl}{avgdl}\right)\right) + tf_{d,t}} \quad (1)$$

$$DF_t = \log\left(\frac{N - df_t + 0.5}{df_t + 0.5}\right) \quad (2)$$

$$QTF_{q,t} = \frac{(k_3 + 1) \times tf_{q,t}}{k_3 + tf_{q,t}} \quad (3)$$

where:

- dl is the document length of d and $avgdl$ is the average documents length.
- $tf_{d,t}$ is the term frequency of t within the document d.
- k_1 is the saturation parameter of $tf_{d,t}$.
- b is the length normalization factor.
- N is the total number of documents in the corpus.
- df_t is the number of documents containing the term t.
- $tf_{q,t}$ is the term frequency of t within the query q.
- k_3 is the saturation parameter of $tf_{q,t}$.

The weight $w_{d,t}$ of a term t within a document d is computed by the $BM25$ weighting function as shown in equation 5:

$$w_{d,t} = TF_{d,t} \times DF_t \quad (4)$$

$$BM25(d, q) = \sum_{t \in q \cap d} w_{d,t} \times QTF_{q,t} \quad (5)$$

The whole BM25 scoring formula is shown in equation 6. We assume that the initial query terms are not weighted, i.e. $\forall q \in Q, \forall t \in q, tf_{q,t} = 1$. Thus, in the PIR models which do not consider the user's profile and exploit only the initial query, $QTF_{q,t}$ can be ignored. In this case, the global IR score of a document for a query is given by equation 5 with $QTF_{q,t} = 1$.

C. User's profile

The user's profile may contain different information types (annotations, comments, citations, social relationships, etc.). In this work, we generate the user's profile from the terms in his annotations.

The user's profile $profile(u)$ is the set of terms which occur within his social annotations:

[2] INitiative for the Evaluation of XML-Retrieval (https://inex.mmci.uni-saarland.de/)

[3] TExt Retrieval Conference (http://trec.nist.gov/)

$$BM25(d,q) = \sum_{t \in q \cap d} \frac{(k_1+1) \times tf_{d,t}}{k_1 \times \left(b - 1 + b \times \left(\frac{dl}{avgdl}\right)\right) + tf_{d,t}} \times \log\left(\frac{N - df_t + 0.5}{df_t + 0.5}\right) \times \frac{(k_3+1) \times tf_{q,t}}{k_3 + tf_{q,t}} \quad (6)$$

$$profile(u) = \{t \in T_z \ / \ a_z = < d, u, T_z > \in A_u\} \quad (7)$$

where A_u is the set of social annotations of u.

The user's profile may contain several occurrences of the same term. Thus we can compute the term frequency $tf_{u,t}$ for a given term t that has been used by u to annotate the documents.

D. $BM25S_{profile}(d,u)$: Personalized Social IR by exploiting only the user's profile

Our aim is to take into account the user's profile during the querying step, in order to personalize the IR process. In $BM25S_{profile}(d,u)$, we replace the query by the user's profile in the scoring function $BM25(d,q)$, in order to retrieve a ranked list of documents that are relevant for a given user u considering only his profile. The query terms are replaced by the user's profile terms: in other words $tf_{q,t}$ is replaced by $tf_{u,t}$ in equation 6.

Unlike the queries, which are composed generally of a few terms, the user's profile contains tens, hundreds or sometimes hundreds of terms. We make the assumption that this implicit user's information need, which is more complex, has to be represented by a vector of term weights, instead of binary weights. We think that a suitable weighting function should distinguish the most important terms from the user's profile. We think also that it should include a saturation effect of the term frequencies, in the same way that the saturation effect of the IR weighting formula BM25 (parameters k_1 and k_3).

Then, we propose three versions of the scoring function $BM25S_{profile}(d,u)$. These scoring functions are computed using the BM25 function (cf. equation 6):

- In the first model $BM25S_{profile-bin}(d,u)$ (binary profile), $k_3 = 0$ in equation 6 corresponds to a maximum level of saturation: the user's profile is represented by a binary vector;
- In the second model $BM25S_{profile-tf}(d,u)$ (term frequencies profile), $k_3 = 1000$ in equation 6 corresponds to set off the saturation level on $tf_{u,t}$ [17]: the user's profile is represented by a vector of term frequencies;
- $BM25S_{profile-w}(d,u)$ (weighted profile), k_3 tuned in equation 6 corresponds to a moderate saturation of $tf_{u,t}$ in order to reach the optimal saturation level: the user's profile is represented by a vector of term weights.

This way, k_3 plays the same role on the user's profile weighting than k_1 on the documents weighting. It remains two differences between terms weighting in the documents and in the user's profile: the first one is that there is no DF component in the user's profile weighting. However, as this component still appear at the document level, it is not useful to use it twice. The second one is that there is no document length normalization in the user's profile weighting, assuming that the users' profiles lengths are more homogeneous than the documents ones.

E. $BM25S_{ScoreComb}(d,q,u)$: Personalized Social IR by combining query and profile at the scoring level

This first combined PIR model, $BM25S_{ScoreComb}$, combines query and user's profile at the scoring level. Given a query q and a user u, for each document d the global score $BM25S_{ScoreComb}(d,q,u)$ is composed of two parts:

- in the first part, on the query side, the classical relevance score $BM25(d,q)$ is computed between the query and a document as detailed in section IV-B;
- in the second part, considering the user's profile side, a PIR relevance score $BM25S_{profile}(d,u)$ is computed between the (non-binary) user's profile and a document as detailed in section IV-D. This PIR score can be computed with one of our three PIR scoring functions $BM25S_{profile-bin}(d,u)$, $BM25S_{profile-tf}(d,u)$ or $BM25S_{profile-w}(d,u)$.

Then, a combined PIR relevance score $BM25S_{ScoreComb}(d,q,u)$ is computed as shown in the equation 8, based on a simple linear combination with the query relevance score and the profile relevance score.

$$\begin{aligned} BM25S_{ScoreComb-X}(d,q,u) &= BM25(d,q) \\ &+ \alpha_1 \times BM25S_{profile-X}(d,u) \end{aligned} \quad (8)$$

where X is "*bin*", "*tf*" or "*w*", referring to one of the three variants of $BM25S_{profile}(d,u)$: $BM25S_{profile-bin}(d,u)$, $BM25S_{profile-tf}(d,u)$ or $BM25S_{profile-w}(d,u)$.

F. $BM25S_{FreqComb}(d,q,u)$: Personalized Social IR by combining query and profile at the term frequencies level

This second combined PIR model, $BM25S_{FreqComb}(d,q,u)$, combines query and user's profile at a terms frequencies level. The user's profile term frequencies are combined with the query term frequencies by a simple linear combination. This leads to replace $QTF_{q,t}$ in equation 4 by the $QTF_{q,u,t}$ defined in equation 9.

$$QTF_{q,u,t} = \frac{(k_3+1) \times [tf_{q,t} + \alpha_2 tf_{u,t}]}{k_3 + (tf_{q,t} + \alpha_2 tf_{u,t})} \quad (9)$$

And the PIR relevance score $BM25S_{FreqComb}(d,q,u)$ of the document d for the query q and the user u is computed with the weighted variant of the BM25 weighting function, as follow:

$$BM25S_{FreqComb}(d,q,u) = \sum_{t \in q \cap d} w_{d,t} \times QTF_{q,u,t} \quad (10)$$

Like in our previous PIR model, we propose three variants of our combined PIR model: $BM25S_{FreqComb-bin}$ with $k_3 = 0$, $BM25S_{FreqComb-tf}$ with $k_3 = 1000$ and $BM25S_{FreqComb-w}$ with k_3 tuned.

V. Experiments and results

Our aim in conducting these experiments is to evaluate, on a Personalized Information Retrieval test collection, PIR models which take in consideration the user's profile. An IR test collection is composed by a set of documents, a set of queries and a set of relevance judgments (i.e. the list of relevant documents for each query). A PIR test collection needs also user-centered data: a set of users asking the queries and a set of user-centered relevance judgments, i.e. the list of relevant documents for each couple (user, query).

A. Personalized IR test collection

With the lack of a PIR test collection providing a list of relevant documents for each query and for each user, we have built a test collection $DelPIR$ based on Web documents and users' annotations extracted from the social collaborative bookmarking network *Delicious*.

We have collected 161,059 documents in 207,304 bookmarks. In *Delicious*, the documents are annotated by the users. So we have been able to collect the 68,600 terms associated to these 161,059 documents by 8,069 users of *Delicious*. We have created automatically 100 queries: each query is composed of 3 terms occurring frequently together in the annotations collected from Delicious. In fact, the 100 triples of terms having the highest Jaccard Index. We have chosen three terms since in practice, most of the time, users formulate their needs with few words [2].

These 100 queries, called "global queries" Q, are issued by 1,118 users, so we obtained a set Q_{UC} of 15,010 couples $(query,\ user)$.

Then we have generated $Qrels$, the set of 107,934 global relevance judgments, i.e. 107,934 couples (q, d) where the document d is relevant considering the query q, independently of the users. A document is globally relevant if it has been annotated by any user with at least 2/3 of the query terms in the same annotation, since three terms seems too restrictive.

And finally, we have generated $Qrels_{UC}$, the set of 137,381 user-centered relevance judgments, i.e. 137,381 triples (q, d, u) where the document d is relevant considering the query q, for the user u. A document is user-relevant if it has been annotated by the user with at least 2/3 of the query terms in the same annotation.

Some statistics about the generated PIR test collection are summarized in the Table II.

TABLE II
PIR test collection data.

$\|D\|$	161,059
$\|U\|$	1,118
$\|Q\|$	100
$\|Qrels\|$	107,934
$\|Q_{UC}\|$	15,010
$\|Qrels_{UC}\|$	137,381

TABLE III
PIR test collection data $DelPIR$ using different filters.

	F_{Doc} filtering	$DelPIR$ F_{Doc} and F_{HQ} filtering
$\|D\|$	161,059	161,059
$\|U\|$	710	443
$\|Q\|$	98	98
$\|Qrels\|$	107,923	107,923
$\|Q_{UC}\|$	6,361	1,363
$\|Qrels_{UC}\|$	93,520	31,910

B. Filtering

We have applied two kind of filters to the PIR test collection:

1) Documents filter (F_{Doc}): we have kept only the couples $(query, user)$ with at least 10 relevant documents ($|qrels_{q,u}| \geq 10$) so that we have obtained 6,361 couples (q, u) in the set Q_{UC}. This led to a reduction of the number of users (710 users left) and thus a reduction of the number of user-centered relevance judgments (93,520).
2) "Hard query" filter (F_{HQ}): we have observed that a lot of the generated queries seem to be very "hard queries", at least for a classical BM25 based IR system. Thus, in order to obtain queries similar to the queries of real users, we have eliminated the most "hard" ones. Using a classical well-tuned BM25 based IR system, we have kept the couples $(query, user)$ obtaining an AP (average precision value) above 0.1%. We have obtained 1,363 couples (q, u) in the set Q_{UC}. This led to a reduction of the number of users (443 users left) and thus a reduction of the number of user-centered relevance judgments (31,910).

Some statistics about the first step of filtering F_{Doc}, and about the final PIR test collection $DelPIR$ obtained after the two steps of filtering F_{Doc} and F_{HQ}, are summarized in the Table III.

The results shown in this section have been obtained by evaluating the classical model and the PIR models on the filtered PIR test collection data $DelPIR$ using F_{Doc} and F_{HQ}. This filtered PIR test collection is composed of 161,059 documents, 443 users, 98 global queries with 107,923 global relevance judgments and 1,363 user-centered queries with 31,910 user-centered relevance judgments. We have evaluated the rankings produced by our PIR models with only user-centered data (Q_{UC} and $Qrels_{UC}$), since the PIR model is not suited to handle global queries.

TABLE IV
$BM25$ EVALUATION RESULTS.

	MAP	$P[0.1]$
$BM25(d,q)$	0.0108	**0.0253**

TABLE V
TUNED BM25 PARAMETERS VALUES.

	MAP		$P[0.1]$	
	b	k	b	k
$BM25(d,q)$	0.6	12	0.6	30

TABLE VI
$BM25S_{profile}$ EVALUATION RESULTS.

	MAP	$P[0.1]$
$BM25(d,q)$	0.0108	**0.0253**
$BM25S_{profile-bin}(d,u)$ with $k_3 = 0$	0.0102	0.0213
$BM25S_{profile-tf}(d,u)$ with $k_3 = 1,000$	0.0093	0.0168
$BM25S_{profile-w}(d,u)$ with tuned k_3	**0.0116**	0.0225

TABLE VII
TUNED BM25 PARAMETERS VALUES.

	MAP		$P[0.1]$	
	b	k	b	k
$BM25S_{profile-bin}(d,u)$ with $k_3 = 0$	0.95	22	0.9	21
$BM25S_{profile-tf}(d,u)$ with $k_3 = 1000$	0.93	23	0.91	21
$BM25S_{profile-w}(d,u)$ Tuned k_3	0.95 $k_3 = 0.08$	22	0.9 $k_3 = 0.07$	21

C. Evaluation measures

We have used the traditional evaluation measures "Precision at 10% of recall" ($P[0.1]$) and "Mean Average Precision" (MAP) which are the most used evaluation measures for evaluating the classical IR systems [18].

D. Evaluation results: $BM25$

The evaluation results of the classical $BM25$ model are shown in Table IV.

In all our experiments we have tuned the parameters of the models (i.e. b and k parameters of BM25) using a grid-based 2D optimization, as proposed by Zaragoza *et al.*, [19]. This well-tuned $BM25$ is a strong baseline for our PIR evaluation described in the next sections.

Table V shows the best values of tuned parameters b and k.

E. Evaluation results: $BM25S_{profile}$

The evaluation results of our PIR model $BM25S_{profile}$ with its variants $BM25S_{profile-bin}$, $BM25S_{profile-tf}$ and $BM25S_{profile-w}$ compared to the baseline $BM25$ are shown in Table VI.

Table VII shows the best values of tuned parameters b and k.

TABLE VIII
$BM25S_{ScoreComb}$ EVALUATION RESULTS.

	MAP	$P[0.1]$
$BM25(d,q)$	0.0108	0.0253
$BM25S_{ScoreComb-bin}(d,q,u)$	**0.0140** $\alpha_1 = 0.13$	**0.0308** $\alpha_1 = 0.13$
$BM25S_{ScoreComb-tf}(d,q,u)$	0.0123 $\alpha_1 = 0.3$	0.0291 $\alpha_1 = 0.58$
$BM25S_{ScoreComb-w}(d,q,u)$	0.0137 $\alpha_1 = 0.26$	0.0306 $\alpha_1 = 0.46$

These results show that the PIR model $BM25S_{profile}(d,u)$ that replaces the user's initial query by his profile, provides better results than the baseline $BM25(d,q)$ when the saturation level is balanced ($BM25S_{profile-w}(d,u)$), considering the MAP evaluation measure: 0.0116 vs 0.0108. The evaluation results of the variants $BM25S_{profile-bin}$, having a maximum saturation level (k_3 = 0), and $BM25S_{profile-tf}$, with no saturation, are worse than the baseline.

Considering the precision at 10% of recall (P[0.1]), the PIR model provides less good results than the baseline.

F. Evaluation results: $BM25S_{ScoreComb}$

The evaluation results of our PIR model $BM25S_{ScoreComb}$ with its variants $BM25S_{ScoreComb-bin}$, $BM25S_{ScoreComb-tf}$ and $BM25S_{ScoreComb-w}$ compared to the baseline $BM25$ are shown in Table VIII.

The results of our first combined PIR model $BM25S_{ScoreComb}$ which combines the classical IR score $BM25(d,q)$ and the social score $BM25S_{profile}(d,u)$, that considers only the matching between the document and the user's profile, are better ($MAP = 0.0140$ and $P[0.1] = 0.0308$) than the results provided by using only the classical model (baseline: $BM25(d,q)$) or only a social score ($BM25S_{profile}(d,u)$).

We observe that, unlike in $BM25S_{profile-bin}$ vs $BM25S_{profile-tf}$ and $BM25S_{profile-w}$, the results of the binary variant $BM25S_{ScoreComb-bin}$ are better than the results of the other two variants ($BM25S_{ScoreComb-tf}$ and $BM25S_{ScoreComb-w}$).

G. Evaluation results: $BM25S_{FreqComb}$

In this section, we present the evaluation results of the PIR model $BM25S_{FreqComb}$ which exploits the user's profile by combining, within the query sent to the PIR system, the initial query terms and the user's profile terms.

The results, shown in Table IX, obtained with $BM25S_{FreqComb}$, are better than the baseline ($BM25(d,q)$: $MAP = 0.0108$ and $P[0.1] = 0.0253$) with all its variant ($BM25S_{FreqComb-bin}$, $BM25S_{FreqComb-tf}$ and $BM25S_{FreqComb-w}$).

We notice that the binary combination $BM25S_{FreqComb-bin}$ having a maximum level of saturation of terms within the query, gives better results ($MAP = 0.0132$ and $P[0.1] = 0.0292$) comparing to the other variants $BM25S_{FreqComb-tf}$ with no saturation and

TABLE IX
$BM25S_{FreqComb}$ EVALUATION RESULTS.

	MAP	$P[0.1]$
$BM25(d,q)$	0.0108	0.0253
$BM25S_{FreqComb-bin}(d,q,u)$ $\alpha_2 = 0.07$	**0.0132**	**0.0292**
$BM25S_{FreqComb-tf}(d,q,u)$ $\alpha_2 = 0.004$	0.0119	0.0258
$BM25S_{FreqComb-w}(d,q,u)$ $\alpha_2 = 0.004$	0.0128	0.0287

$BM25S_{FreqComb-w}$ having a balanced saturation level ($k_3 = 0.07$).

H. Discussion

The results detailed in Table VI and Table VIII highlight two basic statements:

- PIR model which takes into account the user's profile, enhances the relevance score results using the MAP measure which is considered as a global evaluation metric.
- Combining the scores $BM25(d,q)$ and $BM25S_{profile}$ improve the evaluation results comparing to using each score apart.

Moreover, we notice that the PIR model $BM25S_{ScoreComb}$ that combines at the scoring level, the query and the user's informational social profile, gives better results than the PIR model $BM25S_{FreqComb}$ combining the query and the user's profile at the frequencies level:

- $BM25S_{FreqComb}$: $MAP = 0.0132$, $P[0.1] = 0.0292$;
- $BM25S_{ScoreComb}$: $MAP = 0.0140$, $P[0.1] = 0.0308$.

This result does not confirm the conclusion of Robertson *et al.*, according which the "term frequencies level" strategy is better than the "scoring level" strategy [15].

VI. Conclusion

We have presented an approach to build the user's profile from his annotations using a dataset collected from the social bookmarking network *Delicious*. We proposed different ways of considering the occurrences of the important terms for the user within his profile: binary, term frequencies and weighted profile, each corresponding to a different level of saturation. We have investigated then, the different ways of integrating the user's profile.

- Firstly, we have presented a simple PIR model $BM25S_{profile}$ that replaces the initial query terms by the user's profile terms.
- Then, we proposed to combine different scores: the thematic score (classical $BM25S$ score of a document for a query) and the social score $BM25S_{profile}$ within a PIR model we called $BM25S_{ScoreComb}$.
- Afterward, we attempted to combine the initial query terms and the user's profile at the frequencies level, by extending the initial query in our PIR model called $BM25S_{FreqComb}$.

We evaluated our Personalized Information Retrieval models on a PIR test collection we have built using the annotations extracted from *Delicious*. Our preliminary results showed that the PIR models improve the user-centered search (i.e.: for a given query, the list of relevant documents depends on the user) by providing better scores than the baseline.

The PIR model that considers the user's profile, allows to better find the relevant documents for the user than the IR model which does not consider the user's profile.

As future works, we plan to extend the user's profile with further social data including the neighborhood's annotations and experiment different combinations and weightings of the user's profile terms. It could be also interesting to explore the impact of more complex queries composed of more than three terms or based on concepts instead of terms.

VII. Acknowledgement

We would like to acknowledge the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

References

[1] B. J. Jansen, A. Spink, J. Bateman, and T. Saracevic, "Real life information retrieval: A study of user queries on the web," *SIGIR Forum*, vol. 32, no. 1, pp. 5–17, 1998.

[2] J. Kamps, M. Marx, M. de Rijke, and B. Sigurbjörnsson, "Structured queries in XML retrieval," in *Conference on Information and Knowledge Management (CIKM'05)*. New York, NY, USA: ACM, 2005, pp. 4–11.

[3] G. Kazai and A. Trotman, "Users' perspectives on the usefulness of structure for XML information retrieval," in *International Conference on the Theory of Information Retrieval (ICTIR'07)*, 2007, pp. 247–260.

[4] M. Sanderson, "Ambiguous queries: test collections need more sense," in *Conference on Research and Development in Information Retrieval (SIGIR'08)*, 2008, pp. 499–506.

[5] E. Agichtein, E. Brill, and S. T. Dumais, "Improving web search ranking by incorporating user behavior information," in *Conference on Research and Development in Information Retrieval (SIGIR'06)*, 2006, pp. 19–26.

[6] S. M. Kirsch, "Social information retrieval," Ph.D. dissertation, Rheinische Friedrich-Wilhelms-Universitat Bonn, 2005.

[7] J. Diederich and T. Iofciu, "Finding communities of practice from user profiles based on folksonomies," in *Workshop on Building Technology Enhanced Learning solutions for Communities of Practice (TEL-CoPs06)*, 2006.

[8] S. Xu, S. Bao, B. Fei, Z. Su, and Y. Yu, "Exploring folksonomy for personalized search," in *Conference on Research and Development in Information Retrieval (SIGIR'08)*, 2008, pp. 155–162.

[9] Y. Cai and Q. Li, "Personalized search by tag-based user profile and resource profile in collaborative tagging systems," in *Conference on Information and Knowledge Management (CIKM'10)*, 2010, pp. 969–978.

[10] D. Vallet, I. Cantador, and J. M. Jose, "Personalizing web search with folksonomy-based user and document profiles," in *European Conference on Information Retrieval (ECIR'10)*, 2010, pp. 420–431.

[11] A. Kashyap, R. Amini, and V. Hristidis, "Sonetrank: leveraging social networks to personalize search," in *Conference on Information and Knowledge Management (CIKM'12)*, 2012, pp. 2045–2049.

[12] C.-M. Au-Yeung, N. Gibbins, and N. Shadbolt, "A study of user profile generation from folksonomies," in *Workshop on Social Web and Knowledge Management*, 2008.

[13] M. Szomszor, H. Alani, I. Cantador, K. O'Hara, and N. Shadbolt, "Semantic modelling of user interests based on cross-folksonomy analysis," in *International Semantic Web Conference (ISWC'08)*, Berlin, Heidelberg, 2008, pp. 632–648.

[14] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[15] S. Robertson, H. Zaragoza, and M. Taylor, "Simple BM25 extension to multiple weighted fields," in *Conference on Information and Knowledge Management (CIKM'04)*, 2004, pp. 42–49.

[16] S. Robertson, S. Walker, M. Hancock-Beaulieu, M. Gatford, and A. Payne, “Okapi at trec’4,” in *Text REtrieval Conference (TREC’4)*, ser. TREC-4, 1996, pp. 73–96.

[17] K. S. Jones, S. Walker, and S. Robertson, “A probabilistic model of information retrieval: development and comparative experiments, part 2,” *Information Processing and Management*, vol. 36, pp. 809–840, 2000.

[18] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. Cambridge University Press; 1st edition, 2008.

[19] H. Zaragoza, N. Craswell, M. Taylor, S. Saria, and S. Robertson, “Microsoft cambridge at TREC 13: Web and hard tracks,” in *TREC’04*, 2004.